3TU. School for Technological Design, Stan Ackermans Institute offers eleven two-year postgraduate technological designer programmes. This institute is a joint initiative of the three technological universities of the Netherlands: Delft University of Technology, Eindhoven University of Technology and University of Twente. For more information please visit: www.3tu.nl/sai.
3TU.School for Technological Design, Stan Ackermans Institute offers eleven two-year postgraduate technological designer programmes. This institute is a joint initiative of the three technological universities of the Netherlands: Delft University of Technology, Eindhoven University of Technology and University of Twente. For more information please visit: www.3tu.nl/sai.

Software Technology - PDEng projects 2007

5 Piotr Łopatka; Handheld control for wireless sensor networks.
7 Bart Golsteijn; Auto3D Improvement – the beat goes on.
9 Athanasios Chliopanos; Video HW prototyping using COTS programmable hardware.
11 Iulia Dobai; Managing UI complexity of experience demonstrators.
13 Yuchen Zhang; Performance analysis for heterogeneous architectures.
15 Iulia-Hădălina Iota-Vatafu; Monitoring abdominal aortic aneurysm geometries.
17 Charalampos Xanthopoulos; Integration into the national healthcare infrastructure.
19 Jiang Zhang; Mechatronics meta-model for model-driven engineering.
21 Panja Sae-Ul; Feasibility analysis of using ConTest for embedded systems.
23 Lelia Rahman; Ontology-based context-aware Smarties.
25 Chetan Nair; A framework for model-based error detection.
27 Joost Meijs; Eduainment sensor platform application framework.
29 Zan Li; Video Improvement In-A-Box (VI2B).
31 Xing Li; Generic Intermediate Representation for code analysis.
It is a great pleasure to present a glimpse of the wonderful world of projects carried out by the PDEng candidates of the Software Technology programme. During the period from 1 December 2006 to 31 August 2007, a number of PDEng candidates from our programme participated in projects at leading companies and industry-oriented organisations, thereby contributing to the practical application of academic research. This booklet contains a set of concise descriptions of these projects. It shows an overview of the broad spectrum of types of design and development projects our PDEng candidates are involved in. As you can see from these descriptions, our PDEng candidates have contributed to the pre-competitive design and development of technology to improve our quality of life, our lifestyle, and our industry’s productivity and quality. In doing so, they have made a significant contribution to the value of MADE IN HOLLAND.
Challenges
Projects at Philips Research study wireless sensor networks. These networks consist of tiny devices to be used in healthcare and ambient intelligence applications. To speed up the development process, tools are necessary to provide better insight into the behaviour of applications on the sensor nodes. Considering the nature of the sensor networks, handheld devices and radio communication should be used.

Results
A system was designed to give developers monitoring and wireless control of sensor applications using a tablet PC. Analysis showed that it was necessary to separate network applications logically. A service was added to the sensor node that enables easy development of multiple network applications.

Benefits
Application developers can use the implemented system to check the operational properties of the operating system running on the nodes. These properties include clock synchronisation and task information. It is possible to add new properties to the system.

Handheld control for wireless sensor networks

Wireless sensor networks consist of small, spatially distributed autonomous devices with sensing and networking capabilities. Many innovative applications can be built on wireless sensor networks. The potential benefits promoted investigation into healthcare and ambient intelligence applications developed at Philips.

Basic infrastructure for sensor applications
Basic infrastructure is required for building sensor applications. The infrastructure includes the operating system, synchronised communication, power load reduction and other basic system services. These run on the sensor nodes. Development of this infrastructure requires tools for testing. Since sensor nodes are embedded systems, we wanted to test them using radio communication. There are standards and tools for network monitoring and control in the PC/Internet world. However, these systems consume too much power, so they cannot be used for these sensor applications. A new system had to be designed to allow remote monitoring and control of newly developed applications.

Network service to enable multi-programming
Attempting to prototype a system that can monitor a single aspect of the sensor network proved to be a good strategy for problem analysis. This allows difficulties to be identified in the early stage of the project. In our case, challenges appeared to be in the network stack. A solution has been proposed that includes a network service to enable multi-programming, and a monitoring and control application to enable retrieval and control of operational properties on the nodes. The network service is a packet multiplexer and demultiplexer, and can be mapped to the Transport Layer in the ISO/OSI model. The monitoring and control application uses the Remote Procedure Call (RPC) mechanism to retrieve properties from the sensor nodes.
The AutoDJ Improvement project was executed in the Digital Systems & Technologies (DS&T) programme of Philips Applied Technologies. Under the name ‘AutoDJ’, Philips Applied Technologies is developing algorithms, and software implementing these algorithms, to automatically mix songs. That means mixing like a human DJ, not just the simple cross-fades created by modern jukeboxes or software like Winamp. The mixes produced by a human DJ are much more sophisticated. Songs are mixed in and out at the right moment, the beats of the songs to be mixed are synchronised, and the tempo changes gradually from one song to the next. A graphical representation of such a mix between two songs A and B is shown in the figure above.

Producing DJ-like mixes
To be able to produce this kind of mixes, knowledge of the songs to be mixed is required, for example about the tempi, the bars, the metres and the musical keys. Obtaining this knowledge and using it to produce DJ-like mixes in an automated way is a difficult task. At the moment of writing, there is no software available that can perform this task fully automatically and that runs on an embedded system.

A first AutoDJ implementation was available at the start of the AutoDJ Improvement project. However this implementation suffered from a number of problems. These were thoroughly analysed and, based on the results of this analysis, a number of significant improvements have been made to the existing implementation.

Challenges
A first version of AutoDJ was available at the start of this project. As a result of some (functionally driven) design choices, the playback part of that implementation was difficult to understand and maintain, and difficult to change or extend. In addition, the system as a whole was rather inflexible and difficult to use. The goal of the AutoDJ Improvement project was to solve these problems by redesigning and reimplementing part of the system.

Results
The problems with the playback part of the system have been solved by developing a completely new generic system for mixing songs. The usability and flexibility problems have been solved by redesigning and reimplementing the top-level architecture and the public API of the system.

Benefits
A solid foundation has been laid to develop AutoDJ further into a commercial-quality product. A wider range of mixes is now supported, and the mixes can be more complex. A number of limitations of the old system have been removed and the generic nature of the newly developed mix model provides ample opportunities for further experiments and improvements. In addition, redesign and reimplementing the top-level architecture and the public API have made AutoDJ more flexible, easier to change and extend, and yet easier to use and maintain.

Bart Golsteijn
“…Bart’s work has yielded an improved AutoDJ software implementation that will allow Philips Applied Technologies to much better serve our customers.”

P. Dillen, Philips Applied Technologies

AutoDJ Improvement – the beat goes on

The AutoDJ Improvement project was executed in the Digital Systems & Technologies (DS&T) programme of Philips Applied Technologies. Under the name ‘AutoDJ’, Philips Applied Technologies is developing algorithms, and software implementing these algorithms, to automatically mix songs. That means mixing like a human DJ, not just the simple cross-fades created by modern jukeboxes or software like Winamp. The mixes produced by a human DJ are much more sophisticated. Songs are mixed in and out at the right moment, the beats of the songs to be mixed are synchronised, and the tempo changes gradually from one song to the next. A graphical representation of such a mix between two songs A and B is shown in the figure above.
Challenges

Provide insight into the requirements and feasibility aspects of video hardware prototyping within a software-oriented group. Design a re-usable prototyping framework on FPGA that will enable an earlier start of video software development, and therefore earlier time to market. Achieve maximum re-usability of existing components and architectures.

Results

An analysis of the requirements, risks and points of attention that a software group must be aware of to carry out a hardware prototyping project. To gain hands-on experience and to support our claims, we went through all the steps of designing on FPGAs coming up with a proof-of-concept prototype, a hardware emulation infrastructure implemented on an FPGA platform. The framework enables fast integration of new video processing hardware IPs, providing all the adaptation layers required for abstracting from the NXP-specific protocols to the FPGA interfaces and IOs.

Benefits

The developed prototype, although in an immature state, is the first attempt of developing a re-usable and scalable hardware emulation framework within and for the needs of a software group. The group can use it as a base for a powerful FPGA prototyping framework to expand the capabilities of the hardware simulation environment or in similar projects. The documentation produced during this project can also help in decision-making and risk management in future projects. Finally, this project enabled the group to gain valuable knowledge of FPGA prototyping, as well as of tools and methods for IC design and integration.

Dependency between hardware and software development

The video processing software executes on target hardware. This creates a dependency between software and hardware development. The target hardware is not available to the Video Software group in the early stages of development, because it is developed in parallel with the software. To minimise time-to-market, there is a strong preference to complete a major part of the software development before the actual hardware becomes available. Due to the limited resources available on the system, the PC-based simulation framework emphasises the logical and synchronisation aspects of video processing, offering limited debugging capabilities. The solution proposed in this project is a video hardware prototyping framework which enables fast integration of video hardware blocks on an off-the-shelf programmable hardware platform. Due to its high accuracy and resemblance to the final hardware IC, it provides the simulator with the functional video processing aspects which were formerly lacking, thereby enhancing its capabilities.

Flexible prototyping framework

The resulting product, a flexible prototyping framework that enables the rapid prototyping of NXP-compatible video processing modules, is being evaluated both within a hardware simulation environment as well as on-chip testing. The aim is for the video hardware prototype to become the base for a re-usable FPGA prototyping platform used by the Video Software group in future video software development projects.

Athanasios Chliopanos

Video HW prototyping using COTS programmable hardware

“His enthusiasm and drive increased with every new problem encountered. Many problems were encountered, tackled and resolved, taking large development efforts. It paid off, after nine months of hard work we now know better what it takes to realise an FPGA prototype using off-the-shelf hardware.”

K. Brink, NXP Semiconductors

The Nexperia Home Software organisation is responsible for supporting NXP Semiconductors’ sales in the TV application domain by providing cost-effective platform software solutions. The customer base is large and consists of various major TV setmakers. The Video Software group develops software components that implement functions related to video processing and video streaming.
Challenges
The project aimed at providing the stakeholders with a solution that would include a new way to develop experience demonstrators (based on a high-level definition language) in a more intuitive way, with a methodology that would allow greater productivity and easier incorporation of change.

Results
Besides developing an implementation for the proposed solution and two experience demonstrators to prove its feasibility, the project provided insight into the process of specifying user interactions, raised many questions relative to possible other applications of the approach, and answered a few questions relative to its advantages and disadvantages. After considering the advantages and disadvantages, we conclude that the Scenario Engine solution provides good support for rapid application prototyping and better maintainability. From a usability point of view, if the initial learning and understanding phases are overcome, the developers could reach higher productivity and efficiency levels.

Benefits
Prototyping and development of experience demonstrators play a fundamental role in the product development lifecycle. The Scenario Engine solution supports the process by allowing faster demonstrator evaluation of interaction concepts, as well as by delivering a set of compact system requirements.

Managing UI complexity of experience demonstrators

Iulia Dobai

“The result of Iulia’s work is a well documented and working piece of software that definitely will be used in the near future of the Alcor project.”

G.-J. Bloem, Philips Applied Technologies

The ALCOR project at Philips Applied Technologies investigates the feasibility of using new interaction concepts within medical applications by developing experience demonstrators. These experience demonstrators show innovative ways of interacting through input devices (such as the uWand) with 2D images, 2D sets of images or 3D reconstructed volumes of medical images. Fundamental concerns in this prototyping context are speed of development, flexibility of making changes and ease of development.

Developing a rapid application prototyping environment
The ALCOR Scenario Engine project investigated an approach to develop a rapid application prototyping environment for experience demonstrators.

The approach investigated makes use of a high-level definition language, CLIPS and a rule-based reasoning mechanism to infer conclusions (actions) from premises (events and conditions). Through the Scenario Engine approach, the developers express user interactions as a set of rules (in the CLIPS language). During run-time these rules are interpreted and the CLIPS inference engine decides which actions should be invoked for any given input event in the current application context.

This approach was used to develop two experience demonstrators. The first one, even though quite simple from an interaction point of view, proves the feasibility of embedding the Scenario Engine solution in the existing architecture. The second one is a replica of an existing real-life experience demonstrator. A careful comparison of the two experience demonstrators – the original one and the one developed with the Scenario Engine approach – revealed some of the advantages as well as some of the disadvantages of the approach.
Challenges

The initial challenge of this project was to enable performance exploitation and optimisation of an existing image/video processing architecture by analysing and characterising the performance characteristics of the software framework and the underlying platform.

Results

The model of the data flow in the framework shows a problematical design choice in the software architecture, which has been proved by experiments to be the main factor that imposes the bottleneck on the performance. To remove this bottleneck, an improved architecture has been proposed. The performance analysis and exploitation on the platform through a set of macro-experiments provides an inventory of potentials and pitfalls in such a platform that typically determine/disturb the performance of the image/video applications.

Benefits

The performance investigation and characterisation on the framework give a concrete example for analysing performance on the image/video processing systems. Such experience can be useful for performance analysis of other applications, typically for applications with a similar architecture. The knowledge and findings from the platform exploitation can help the image/video algorithm developers to estimate the performance of their applications, and should be taken into account in making performance relevant design decisions. At the organisational level, the resulting knowledge and findings contribute to the database of platform knowledge, which is crucial for projects to develop performance-aware real-time image/video applications.

In the Video Signal Processing group of Philips Applied Technologies, a video processing framework has been developed on the PC platform that deploys a pipelined architecture on the CPU-GPU base. This provides flexibility and extendability to accommodate different algorithms to process various formats of image/video data. One algorithm that has been realised in this framework is Natural Motion, which produces a smooth impression of moving objects in videos by producing the intermediate frames between two original frames in real time. There is growing interest in applying this framework to real-time image reconstruction algorithms in professional medical scanners, such as X-ray, computer tomography and ultrasound. However a critical question to be addressed is whether running an algorithm on such a framework can guarantee sufficient and predictable performance to meet the requirements of specific applications with respect to throughput and latency.

Methodology and techniques for modelling data flow efficiency

The performance investigation and characterisation on the framework gives a concrete example for analysing performance on the image/video processing systems. More particularly it concerns the methodology and techniques for modelling the data flow efficiency in such a system, detecting the latency/throughput bottlenecks, and assessing and improving the architecture from the performance viewpoint. The model of the data flow in the framework shows a problematical design choice in the architecture, which has been proved by experiments to be the main factor that limits performance. An improved architecture has been proposed to remove this bottleneck. The performance analysis and exploitation on the PC platform by means of a set of macro-experiments provided a list of potentials and pitfalls in such a platform that typically determine/disturb the performance of GPU-based image/video applications. The identified knowledge and findings can help the image/video algorithm developers to estimate the performance of their applications, and should also be taken into account by the developers in making performance-relevant design decisions.
Challenges
One of the challenges was how to create a solution that accurately helps clinicians in easily and automatically diagnosing AAA diseases. To achieve this, parts of existing software had to be re-used. This raised another challenge: how to build the new software on the existing one.

Results
The result of the project consists of a prototype which measures and monitors the AAA geometry based on 3D CTA and MRI data. The tests performed on the prototype show that the computation of the geometrical parameters (which describe the AAA geometry) is accurate. In addition, recommendations are made for further improvement of the prototype to allow a useful solution to be offered to clinicians.

Benefits
Before introducing a product to the market, a prototype should first be clinically evaluated. The developed prototype gives Philips Medical Systems the possibility to show clinicians an application which measures and monitors the AAA geometry over time, based on MRI and CTA data. A clinical evaluation will then show to what extent the prototype meets the clinicians’ needs.

MeMo – Measurement and Monitoring of abdominal aortic aneurysm geometry using 3D CTA and MRI
Abdominal aortic aneurysm (AAA) is a relatively common vascular disease and has a significant impact on peoples’ lives. The aortic aneurysm is one of the main causes of death, mainly in men aged over 55. The rupture of an aneurysm usually causes immediate death. Timely identification of patients at risk of AAA rupture can lead to better management of the disease.

Improving diagnosis of vascular diseases
One of the current goals of the Healthcare Informatics business unit of Philips Medical Systems is to improve the diagnosis of vascular diseases, including AAA disease. At present, an AAA is regularly monitored using two-dimensional (2D) ultrasound imaging. However the accuracy of ultrasound imaging is limited. During the last decade, the three-dimensional (3D) imaging techniques Computed Tomographic Angiography (CTA) and Magnetic Resonance Imaging (MRI) have become available. These techniques can allow more accurate measurement of AAs.

To achieve the goal of improving the diagnosis of an AAA, a prototype AAA geometry measurement and monitoring tool was designed and implemented. As a proof of concept, a prototype was created which performs landmark detection (i.e. finding characteristic positions for comparing the AAA geometries), computation of geometrical parameters, comparison of parameters for two or more AAs and the visualisation and reporting of the AAA geometries and associated results.

Measuring and monitoring AAA geometry
The prototype as developed provides a basis for further investigation to improve measurement of the AAA geometry. It therefore gives Philips Medical Systems the possibility to show clinicians an application which measures and monitors the AAA geometry over time, based on MRI and CTA data. A clinical evaluation will have to show to what extent the prototype satisfies the clinicians’ needs. After successful clinical validation, the prototype may be further developed into a product.

Iulia-Mădălina Ioța-Vățăfu

Monitoring abdominal aortic aneurysm geometries

“The prototype software application that Mada has developed... perfectly matches our expectations! The prototype forms an excellent basis for future clinical evaluation and for potential future product development.”

M. Breeuwer, Philips Medical Systems
Challenges
The major challenge we had to deal with stemmed from the considerable fluidity and vast diversity pervading the establishment of interoperability in healthcare. Official specifications for connection to the healthcare IT infrastructure are either incomplete or still undergoing a long process of maturing. The two countries we examined—Germany and the Netherlands—although neighbouring, have applied different standards and technologies in their national healthcare infrastructures. Furthermore, standardisation committees issue their guidelines, while groups of experts are keen to promote their own solutions. In the light of these developments, the underlying challenge proved to be the derivation of an architecture that would enable connectivity with the national healthcare infrastructures, while maintaining a loose coupling with their intrinsic features.

Results
The results of the project include an extensive documented analysis of interoperability in the healthcare domain, highlighting the various alternatives, along with a proposed architecture and design. Furthermore, we developed a prototype to show the creation and transmission of a referral note to the Catharina Hospital, Eindhoven, and the reception of the corresponding discharge note, in accordance with the requirements of the Dutch healthcare infrastructure.

Benefits
Philips Applied Technologies—Digital Systems & Technologies has enriched its information inventory with knowledge about the status, trends, and alternatives in the domain of interoperability in healthcare. It also has a tangible demonstrator in cooperation with a prestigious healthcare institute, the Catharina Hospital, which it can use for promotional purposes to potential customers.

Integration into the national healthcare infrastructure

Integrating Philips Motiva into the Dutch and German national healthcare infrastructures
In recent studies, home telemonitoring has been proven to reduce mortality and hospitalisation rates for chronic heart failure patients. Philips’ remote patient management technology enables care teams to monitor patients’ progress outside the hospital, promoting patient self-management, as well as facilitating two-way communication between patients and their care providers.

Engaging patients with personalised healthcare content
Philips has developed Motiva, a new TV-based telemedicine platform, designed to engage patients with daily, personalised healthcare content, and to help care managers reach more patients, influence long-term behaviour change and reduce healthcare costs. Philips Motiva took the Medical Device & Diagnostic Industry by storm, winning prestigious awards and being praised by healthcare consortiums in this field. Meanwhile, demands for the formulation of a set of standards to enable information exchange between healthcare providers in different countries were increasing. Most notably in Germany and the Netherlands, specific governmental institutes took up the challenge of developing a suitable framework for establishing interoperability in the healthcare domain. The objective was to define a set of protocols that could standardise the format of the exchanged information, along with an architecture that could support the interconnection of the various healthcare providers. "The work of Charalampos resulted in..., and better understanding of the domain, not only for him but also for me and probably others as well.”
D. La Hei, Philips Applied Technologies

Integration into national healthcare infrastructures
Philips Applied Technologies foresee an interesting business opportunity that was worth exploring. It envisaged integrating the Philips Motiva system into the national healthcare infrastructures. One of the first steps was therefore to stipulate a number of scenarios for further investigation and prototyping, such as the referral of a Motiva patient to a doctor and the subsequent discharge from the hospital. The developed demonstrator combines multiple technologies such as Web Services, SOAP, XML, Java and the Health Level 7 Version 3 protocol, a widely accepted protocol for modelling workflows, entities and concepts to enable interoperability in healthcare. To enhance the persuasive power of the demonstrator, we launched a joint venture with the Catharina Hospital Eindhoven to visualise the exchange of information in a real healthcare IT environment.

Charalampos Xanthopoulos

“The work of Charalampos resulted in..., and better understanding of the domain, not only for him but also for me and probably others as well.”
Challenges
The project was motivated by the need for highly dynamic, rapidly reconfigurable, domain-specific modelling of mechatronics applications. Having developed successful systems in the past, Philips Applied Technologies wanted to find an efficient and effective way of developing and modelling new mechatronics systems by re-using components of legacy products, even in binary form. In addition, Philips Applied Technologies wanted to have a domain-specific language and environment for the modelling of component-based mechatronics applications.

Results
The project resulted in a domain metamodel, which consists of premodeled mechatronics components that contain domain knowledge. The domain metamodel allows the architect to design applications by re-using and integrating components identified from legacy applications. The project also involved the design and development of a new graphical modelling language and environment, which includes new notation and semantics specifically for the modelling of component-based applications in the mechatronics domain.

Benefits
MDE raises the level of abstraction in all aspects of software development throughout its lifecycle by making models primary artifacts. Our component-based MDE approach especially brings benefits such as the division of labour, because components can be predefined by one or a few domain experts. In addition, the design models made by our tools can generate codes for deploying and integrating binary-form legacy components. That significantly improves developer productivity and code quality. The new domain-specific modelling language supports higher-level abstractions than General-purpose MDE, and requires less effort and fewer low-level details to design an application in the domain in question.

Mechatronics metamodel for Model-Driven Engineering

"It is clear that not many developers would be capable of bringing such a project to a successful end. We are therefore very grateful to Jiang Zhang for the work that he has done..."

M. Beekveld,
Philips Applied Technologies

The project involved the design of a component-based metamodel that could be used for domain-specific Model-Driven Engineering (MDE). MDE holds great promise for the future of software engineering. In this research project, Philips Applied Technologies wanted to investigate the applicability of MDE to the software development of motion control applications with the Equipment Control Platform (EqCP), which is a real-time software platform for the component-based development of mechatronics applications. Philips Applied Technologies’ employees and existing motion control software provided the input for the design of this metamodel.

Designing motion control applications
The MDE metamodel allows the modeller to design Motion Control applications by integrating the premodeled mechatronics components which have been identified from legacy applications. Both the metamodel and design models were made with the EqCP Modelling Environment (EME), which is a graphical modelling environment designed and developed in this project.

In particular, the project produced a workable mechatronics metamodel which could be used for modelling of motion control applications and generating code. Finally, a case system of motion control was developed with the proposed MDE approach and metamodels, to show the applicability of MDE to mechatronics software development.

This project succeeded in putting the MDE and metamodelling theory and principles into practice, and in investigating the technical feasibility of and techniques for applying MDE to the domain-specific component-based software development of mechatronics applications.
Challenges

The challenge was to integrate ConTest into a real-time embedded system. The characteristics of embedded systems are very different from those of enterprise software systems like IBM WebSphere®. The integration of ConTest into the TV software therefore included a number of modifications to port ConTest to the resource-constrained and platform-specific TV environment. As the program timing of a real-time system is designed accurately for its limited resources, it can also be problematic to add an additional program to run in such a system without impairing its real-time properties. Moreover, the fact that our case study was not a small real-time system but a large industrial system made the project even more challenging.

Results

C-ConTest was successfully integrated into the TV software stack. Based on the current evaluation results, C-ConTest still has many points of improvement in the real-time embedded system domain in terms of its effectiveness in revealing and eliminating concurrency bugs. Moreover, although the implementation of C-ConTest adheres to the POSIX standard, this does not mean much in the real-time embedded system domain where each company is likely to have its own predefined basic functionalities. C-ConTest cannot assume what resources would be available in the target system.

Benefits

This project made the technical issues in integrating a test tooling into the TV software stack clear to Philips Consumer Electronics (Philips CE). The same integration approach can be used when Philips CE wants to integrate other tooling into its software. The project also allowed IBM to gain more insight into technical issues in the real-time embedded system domain, and to improve ConTest for this domain. Moreover, the project established the cooperation between Philips CE and IBM. In the near future, Philips CE can expect improvements of ConTest in many aspects, and it could turn out to be a potential concurrency-test tooling for Philips CE in the future.

Panja Sae-Ui

Feasibility analysis of using ConTest for embedded systems

Testing a multi-threaded system is difficult and costly. Because tasks in a multi-threaded system can interact with each other and the space of possible execution paths is extremely large, multiple executions of the same test may have different interleavings and different results. Moreover, when a concurrency problem is discovered it is very hard to reproduce and debug. As software is becoming more and more complex, Philips CE foresees difficulty in tackling concurrency problems in its TV software in the future, and wants to investigate better ways to deal with these problems.

Forcing concurrency-related bugs to appear

ConTest, a concurrency test tooling by IBM’s Verification and Testing Solutions group in Haifa, provides a means to force concurrency-related bugs in multi-threaded systems to appear with high frequency. It systematically and transparently schedules the execution of programme threads into scenarios where race conditions, deadlocks and other intermittent bugs are likely to occur. This helps to expose concurrency problems early in the testing process. ConTest improves the quality of testing and reduces development costs. ConTest has been used successfully on industrial-size applications (such as IBM WebSphere®). However, it has never been used in real-time embedded systems. The objective of this project was to investigate the effectiveness of ConTest in this domain, using a TV software stack provided by Philips CE as a case study.

Successful integration into the TV software stack

C-ConTest, the C version of ConTest, was used in this project. It is still under development and only works with the pthread library while the TV software stack uses VxWorks primitives to achieve concurrency. In this project, one of the main challenges was to port C-ConTest from the pthread library to VxWorks. Moreover, C-ConTest was split into two explicit packages: ConTest instrumentor and ConTest run-time library. Only the latter was integrated into the TV platform. This approach kept the part running on the TV platform as small as possible to avoid integration problems. In this project, C-ConTest was successfully integrated into the TV software stack. A number of technical issues were identified and solved during the integration phase. The evaluation of the project showed that C-ConTest still has many points of improvement in the real-time embedded system domain. Recommendations were given to IBM for future works.

"His drive for results and solid way of working kept the project on track... the technical work was done in a professional way. No cutting corners to 'get it working', but creating production-quality implementations."

C. Aarts,
Philips Consumer Electronics
Challenges
Extending the architecture of devices that can collaborate to fulfill specific user requests by building an ad hoc network of functionalities. The extension includes features such as learning capabilities between the devices, and context-awareness. The architecture should be validated by a working prototype.

Results
• Architecture of collaborative devices that can learn from each other and are context-aware.
• Knowledge base design and generic software design that fulfill the architecture.
• A prototype proof of concept that implements this architecture using UPnP technology and development boards based on PNX0106.

Benefits
The result of this project was a proof of a new concept for semiconductor devices, in which these devices can autonomously collaborate to fulfill user requests. The prototype showed that these devices can auto-configure themselves as new devices come along, enabling the user to evolve the network by buying small devices which add new functionalities to the network. The prototype also showed the feasibility of context-awareness in the concept.

The Advanced Systems Laboratory (ASL) within NXP is a research group which looks three to four years ahead with the aim of developing concepts for future devices. ASL rapid prototypes these concepts and tests them with prospective customers, at the same time finding out how NXP chips can be improved to meet these concepts. As a step towards the Ambient Intelligence vision, ASL has developed an ad hoc network concept called Collaborative Smarties, using a new type of device called a Smarty. A Smarty provides a set of functionalities and is able to participate in forming a virtual device in an ad hoc way. A virtual device provides the user with a combination of functionalities embedded in different physical Smarties as if it were a single device.

Developing the Collaborative Smarties concept
In this project, we would like to develop the Collaborative Smarties concept on a number of aspects. For example, when the user buys a new device, the existing Smarty devices (Smarties) should learn about this new device and acquire the capability of forming an ad hoc network using this new device. Another aspect is that the Smarties should recognize the different contexts in which people live, such as relaxing, sleeping, eating, partying, and so on. In other words, Smarties should fulfill a specific user request according to the context the user is in at that time.

We have extended the concept of Collaborative Smarties to address the above issues of the Smarty technology. By embedding ontology-based knowledge, we showed that Smarties can auto-configure themselves as each new Smarty comes along. We also showed that each Smarty can react differently to different contexts. Updatability of the knowledge base enables devices to learn new contexts from different devices.

Ontology-based context-aware Smarties

“Leila has contributed significantly towards the goal of realizing a network of autonomous devices. We are very happy with the results, and further work will be undertaken based on Leila’s results.”

A. Syed, NXP Semiconductors

Leila Rahman

Leila has contributed significantly towards the goal of realizing a network of autonomous devices. We are very happy with the results, and further work will be undertaken based on Leila’s results.”

A. Syed, NXP Semiconductors

Ontology-based context-aware Smarties

The Advanced Systems Laboratory (ASL) within NXP is a research group which looks three to four years ahead with the aim of developing concepts for future devices. ASL rapid prototypes these concepts and tests them with prospective customers, at the same time finding out how NXP chips can be improved to meet these concepts. As a step towards the Ambient Intelligence vision, ASL has developed an ad hoc network concept called Collaborative Smarties, using a new type of device called a Smarty. A Smarty provides a set of functionalities and is able to participate in forming a virtual device in an ad hoc way. A virtual device provides the user with a combination of functionalities embedded in different physical Smarties as if it were a single device.

Developing the Collaborative Smarties concept
In this project, we would like to develop the Collaborative Smarties concept on a number of aspects. For example, when the user buys a new device, the existing Smarty devices (Smarties) should learn about this new device and acquire the capability of forming an ad hoc network using this new device. Another aspect is that the Smarties should recognize the different contexts in which people live, such as relaxing, sleeping, eating, partying and so on. In other words, Smarties should fulfill a specific user request according to the context the user is in at that time.

We have extended the concept of Collaborative Smarties to address the above issues of the Smarty technology. By embedding ontology-based knowledge, we showed that Smarties can auto-configure themselves as each new Smarty comes along. We also showed that each Smarty can react differently to different contexts. Updatability of the knowledge base enables devices to learn new contexts from different devices.
Challenges
A model-based framework for awareness in an embedded system may only have minimal impact on the existing (possibly real-time) behaviour of the system. This imposes many design constraints, for example limited processing power, memory and bandwidth. Another challenge is the accuracy of error detection, which is influenced by factors such as correctness of implementation of the model, proper timing of measurement and comparison of behaviour, and variability in the acceptable behaviour of the system (determining what is a deviation and what is not).

Results
A framework for model-based awareness was designed. It allows quick experimentation with different models, and is configurable to suit different strategies for detecting and reporting errors. An Awareness Monitor was implemented on Linux using this framework, and was shown at the TRADER Demo Day. TRADER research partners are using the framework in experiments in monitoring MPlayer (a popular open-source media player application). The Awareness Monitor can be ported to embedded platforms such as a TV with relatively low effort.

Benefits
The researchers at ESI now have a framework available for conducting experiments on model-based awareness. They have increased knowledge of topics in model-based awareness such as modelling behaviour and detecting and reporting errors. The research partners of TRADER are using the framework to aid their research. Trials of model-based awareness on embedded systems such as TVs can be started using the framework. The framework is easily used for black box testing for test-time improvements in reliability.

Reliability is a prime concern in software-intensive embedded systems such as those found in TVs and printers. Trends in embedded systems design such as increasing complexity and shortening product life cycles demand better development methods to maintain product reliability at the required levels. The TRADER project at the Embedded Systems Institute (ESI) aims to develop novel methods to ensure the reliability of consumer electronics products.

Model-based awareness: promising approach to run-time reliability improvements
Model-based awareness is a promising approach to run-time reliability improvements. Model-based awareness makes a system aware of its own behaviour using a model. A behavioural model specifies the desired behaviour of a system. A deviation in the system from its specified behaviour must be detected as an error. An Awareness Monitor uses the model to detect errors in the system under observation. Subsequently, the awareness monitor initiates a diagnostics and recovery component to locate and recover the erroneous parts of the system under observation from errors. As a result, a system that is not technically error-free appears reliable to its users. ESI researchers investigating model-based awareness have created behavioural models of embedded systems (e.g. in a TV) in Stateflow (a design and simulation tool for event-driven systems). They needed a framework to experiment with model-based awareness on these systems.

Developing an awareness framework
In this project, an awareness framework was developed to introduce awareness into systems using behavioural models in Stateflow. Based on this framework, an Awareness Monitor was implemented on Linux to monitor a Linux process. The framework is easily adapted to use different models, enabling quick experimentation with various models. The framework is configurable to suit different strategies for detecting and reporting errors. The system under observation requires minor modifications to allow reporting relevant measurements to the framework. Although extensive measurements were not performed, initial performance measurement results indicate that the load the Awareness Monitor places on the system is relatively low.

“Chetan’s work has also triggered new research on the simulation of the awareness framework: we expect substantial impact, both scientifically and industrially, in the near future.”

T. Hendriks, Embedded Systems Institute

Chetan Nair

A framework for model-based error detection
Challenges
Creating a framework that supports the creation of tangible interaction consoles and enabling the creation of different applications for those consoles was a major challenge. The framework combines a simple run-time environment with the possibility of specifying complex behaviour, while at the same time offering flexibility and ease of use, all in one system. In addition, embedded hardware constraints had to be met to allow the run-time environment to be deployed.

Results
A framework for fast, easy creation and execution of autonomous tangible sensor-based applications was created. The framework enables the specification of tangible interaction consoles, and forms a basis on which different types of tailored user interfaces can be built, to allow a variety of users (from children to manufacturers) to create applications for the consoles.

Benefits
The framework is considered to be an important step towards moving the creation of autonomous tangible sensor-based applications from technologically savvy people to those who are more oriented towards toy and game design. In addition, the framework speeds and simplifies the creation of tangible sensor-based applications for all types of users.

Edutainment sensor platform application framework
The project was carried out at Philips Research, where sensor-based tangible user interfaces have been studied since early 2003. The first demonstration of a tangible user interface was made in the form of a story-telling environment. Along the way a wider applicability of the platform has been recognised. This platform is now known as the Edutainment Sensor Platform or ESP.

The first target of ESP is educational games for children. Next to that, ESP aims at a wide variety of other applications such as physical therapy, elderly gaming and home control. The ESP enables the creation of these applications while hiding the underlying technology. The technology behind ESP consists of different (wireless) sensors, actuators, embedded computing and development tools. The key feature of the platform is ease of use.

Fast, easy creation of autonomous tangible applications
The objective behind the ESP Application Framework is to enable fast, easy creation of autonomous tangible applications by overcoming two problems. The first of these problems is the need of low-level sensor programming to create an application, which is time-consuming. The second problem is that the people who are capable of programming the sensors mostly do not possess the creative skills needed to build ‘fun’ applications. By creating a framework that overcomes these two problems, the creation of sensor applications is greatly simplified and becomes possible for people who are less technologically savvy and more design-oriented.

Those problems are tackled by splitting the system into two parts: design and run-time environment. The design environment enables the user to create an application; subsequently the user can deploy the application to the run time environment. The run time environment interacts with the physical world according to the application created on the run time environment. To interact with the physical world a number of sensors and actuators are connected to the (embedded) run time environment. A specific set is known as a tangible interaction console. To facilitate the creation of applications by different types of users, the system incorporates a dedicated programming language (ESPranto) and a flexible hardware abstraction layer.

To validate the system, two demonstrators (StoryToy and TagTiles) have successfully been (re)created. StoryToy is a single-player story-telling game (slow and simple), and TagTiles is a two-player pattern-matching game (fast and complex). The demonstrators show that the framework allows the creation of autonomous tangible applications.

Joost Meijles
Joost did his work diligently and following an organised and well documented process. It gives us the evidence and confidence that the first version of the platform is already a good and useful one. In all, a formidable job well done.”

Willem Fontijn, Philips Research
Challenges
There were three main challenges:
• Exploration of state-of-the-art video processing algorithms which should not only give pleasing visual results but also should be commercially feasible for use in the product.
• Real-time constraints in software development.
• Analyzing and realizing the synchronization mechanism.

Results
A prototype that contains three software video processing algorithms was designed and implemented. A flexible architecture which defines clear video algorithms, interfaces and control hardware on the board was successfully implemented.

Benefits
The VI2B delivered the software which is able to control the hardware and integrate three new advanced video processing algorithms. The buffer manager scheme is designed and implemented. One of the processing algorithms gives impressive results. This is probably commercially viable for use in the product if the computation complexity demands on the hardware are greatly reduced.

---

Zan Li

**Video Improvement In-A-Box (VI2B)**

“Zan Li was able to show her skills in (advanced) video processing, and after one weekend of hard work she showed the first results of her implementation of an edge-directed upscaling algorithm.”

K. Brink, NXP Semiconductors

NXP Semiconductors, as one of the world’s larger semiconductor companies, provides a variety of integrated circuits (ICs) for home, identification, mobile, automotive and multimarket semiconductors. The video group of the Home Innovation Center Eindhoven provides the video competence to the business line digital TV system of NXP’s Home business unit.

**Significant differences in video source quality**

The diversity of video sources has increased tremendously in recent years, with significant differences in their quality (resolution, frame rate, encoding accuracy). At the same time, display devices such as TVs have increased in size, resolution and reproduction quality. Typically, today’s TV systems with these high resolution displays are perceived as underperforming when rendering video from low resolution input, low quality video sources. This often results in disappointed customers, who tend to blame the display for the perceived low quality.

Nowadays, the key element which makes a TV competitive on the market is the video improvement. This project aims at bridging the gap between the poor video quality of low-resolution sources and high-resolution displays by developing a device called Video Improvement In-A-Box (VI2B). This is based on existing state-of-the-art video software algorithms and video hardware available in the Video group of NXP Semiconductors.

**Flexible architecture which controls the hardware**

The VI2B application must deal with real-time constraints and synchronization between multiple threads. The proposed solution defines a flexible architecture which controls the hardware. This architecture defines clear software algorithm interfaces that can be used to improve visual quality. A proof of prototype has been designed and implemented to show the feasibility of the architecture.
Challenges
One of the challenges of this project was to define a representation of a programming language which acts as an intermediate media in projects of code analysis in LaQuSo. This representation should be generic so that it is feasible to map any programming language into it, and the metrics and the diagrams that LaQuSo needs can be extracted from it. Another challenge was to generalise the mapping from specific language(s) to the defined representation, as well as metrics and diagram extraction from the representation so that developers will have guidelines to map another language to the representation and extract other kinds of metrics or diagrams from it. In addition, the extraction system should be re-used in the projects of code analysis for different languages.

Results
The Graph eXchange Language (GXL) format was used as the generic representation according to LaQuSo requirements. Furthermore, a flexible metamodel was defined to accommodate a broad range of programming languages and metrics. Two software projects, which were written individually in Java and MicroFocus COBOL, were successfully mapped into GXL. Additionally, 13 kinds of metrics ranging from the number of operators to Depth of Inheritance Tree (DIT) were extracted from the GXL representation. The extracted metrics were verified. Experiments showed that the extraction system is re-usable. Above all, the mapping and extraction guidelines were also proposed.

Benefits
Given the fact that as long as a programming language can be mapped to GXL, a uniform way can be employed to extract metrics and diagrams, the time spent on code analysis for different languages can be reduced with respect to the metrics and diagram extraction. Moreover, if GXL is used with a re-usable extraction system to facilitate code analysis, the only effort that is in principle required will focus on mapping a programming language to the representation. The most important goal of this project was to define such a generic representation for code analysis, from which diagrams and metrics are extracted in a united way. The work of extraction is intended to be re-used for different languages. The time spent on code analysis for different languages can therefore be saved with respect to metrics and diagram extraction. Moreover, if a generic representation together with a re-usable extraction system is used to facilitate code analysis, the only effort that is in principle required will focus on mapping a programming language to the representation. The most important goal of this project was to define such a generic intermediate representation for any programming language. To achieve this, we compared and evaluated similar subjects from multiple research groups. Once it was defined, two software projects, written in Java and MicroFocus COBOL, were used as case studies. By mapping the two software projects to the defined intermediate representation and extracting metrics from the mapping results, we have generalised the mapping and extraction processes to obtain guidelines for other developers. In addition, a re-usable extraction system was also designed during the extraction of metrics.

Generic intermediate representation for code analysis

This project was performed in the Laboratory for Quality Software (LaQuSo), which helps customers to analyse the quality of their software code. To do that, metrics and diagrams are extracted from the code. Diagrams refer to class, sequence, activity diagrams etc., which are defined by Object Management Group. Metrics are normally statistics, such as the number of ‘if’ statements in a programme. Based on metrics and diagrams, the quality of software is judged. The software that customers have is written in different languages, for instance C++ and COBOL. What LaQuSo does for them is: first, parsing the code to get a syntax tree, which can also be regarded as an intermediate representation of a language. But it is always bound to a specific language, in other words a different language has different syntax tree. Then, based on the syntax tree, language-specific methods are built to extract metrics and diagrams. The problem in this process is that the work spent on extracting metrics and diagrams for different languages is repeated, and sometimes even overlaps.

Defining a generic intermediate representation
To tackle that problem, researchers at LaQuSo advocate the idea of defining a generic intermediate representation, from which diagrams and metrics are extracted in a united way. The work of extraction is intended to be re-used for different languages. The time spent on code analysis for different languages can therefore be saved with respect to metrics and diagram extraction. Moreover, if a generic representation together with a re-usable extraction system is used to facilitate code analysis, the only effort that is in principle required will focus on mapping a programming language to the representation.
Edited by:
H.J.G. Weffers
Coordinator Software Technology
Eindhoven University of Technology

Text:
PDEng candidates of the Software Technology programme

Production:
Lian van Hout communicatie, Eindhoven

Translation:
Nigel Morrison, Eindhoven

Photography:
Rien Meulman, Eindhoven

Design:
vanRixtelvanderPut ontwerpers, Eindhoven

Printing:
Drukkerij Lecturis, Eindhoven

Piotr Łopatka; Handheld control for wireless sensor networks.
Bart Golsteijn; AutoDJ Improvement – the beat goes on.